A heuristic for the skiving and cutting stock problem in paper and plastic film industries

Yan Chen[1, 3], Xiang Song[2], Djamila Ouelhadj[2], Yaodong Cui[3*]

1 School of Business Administration, South China University of Technology, Guangzhou 510640, China

2 Department of Mathematics, University of Portsmouth, Portsmouth PO1 3HF, United Kingdom

3 School of Computer, Electronics and Information, Guangxi University, Nanning 530004, China

Abstract: This paper investigates the skiving and cutting stock problem (SCSP) encountered in the paper and plastic film industries, in which a set of non-standard reels generated from previous cutting processes are used to produce finished rolls through the skiving and cutting process. First, reels are skived together length-wise to form a reel-pyramid (a polygon) and then, the reel-pyramid is cut into finished rolls of small widths. Depending on if a reel can be divided length-wise into sub-reels to form the reel-pyramid, the problem can be classified into divisible SCSP (DSCSP) and indivisible SCSP (ISCSP). In this paper, two integer programming (IP) models are proposed for DSCSP and ISCSP respectively. A sequential value correction procedure combined with the two IP models (SVCTIP) is developed to solve the two SCSPs. The effectiveness of the SVCTIP is demonstrated though extensive computational tests.

Keywords: Combinatorial optimization; Skiving and cutting stock; Reel cutting; Cutting problems

1. Introduction

Various types of cutting stock problems have been investigated in the literature (Beeker and Appa, 2015; Wei et al., 2016; Arbib et al., 2016; Garraffa et al., 2016). Effective algorithms for solving them are useful to improve material utilization and reduce production cost.

The skiving and cutting stock problem (SCSP) investigated in this paper is encountered in the paper and plastic film industries. In SCSP, there is a set of non-standard reels generated from previous cutting processes, including past leftovers of cutting patterns, over-makes, and salvaging to remove defects, etc. These reels are rectangles with disparate lengths and widths. They can be used to satisfy customer orders for rolls (rectangle shape) with an objective of using the minimum total area of the reels to produce the required rolls.

[*] Corresponding author

Email: ydcui@263.net

Because the reel lengths are generally smaller than the roll lengths, the skiving and cutting process is used to produce finished rolls from the reels. It consists of the skiving stage and the cutting stage. In the skiving stage, the reels are skived together length-wise to form a reel-pyramid (a polygon) and in the cutting stage, the reel-pyramid is cut into finished rolls of small widths. An SCSP example is provided to illustrate the skiving and cutting process in details in the following paragraph. In the SCSP, the widths of reels/rolls are measured in millimetres and the lengths in meters; the length of a reel/roll is much larger than its width. For the convenience of presenting the skiving and cutting process, different scales are used for the width (horizontal) and length (vertical) directions in the figures of the rest of the paper.

Let W_j, L_j and N_j be the width, length and number of available reels of type-j, $j = 1, \cdots, n$, where n is the total number of reel types. Then $D_j = N_j L_j$ is the total length of type-j reel. In the SCSP example, $n = 7$ and other reel data are provided in Table 1. In industry, the number of reel types is often in the range of [5, 30].

Table 1. Reel data of the SCSP example

j	1	2	3	4	5	6	7
W_j	1215	1200	1180	1130	1120	960	920
L_j	4500	3000	3000	3000	3200	5000	5000
N_j	1	1	1	1	1	2	2
D_j	4500	3000	3000	3000	3200	10000	10000

Let w_i, l_i and d_i be the width, length and demand of type-i roll, $i = 1, \cdots, m$, where m is the total number of the roll types. In the SCSP example, $m = 5$ and other roll data are provided in Table 2.

Table 2. Roll data of the SCSP example

i	1	2	3	4	5
w_i	260	310	320	350	405
l_i	10000	10000	10000	6000	6000
d_i	2	3	2	2	1

Depending on if a reel can be divided length-wise into sub-reels to form the reel-pyramid, the problem can be classified into divisible SCSP (DSCSP) and indivisible SCSP (ISCSP).

The solution of the ISCSP is exactly one cutting plan that contains exactly one reel-pyramid and one roll-pyramid. The reel-pyramid (see Figure 1) is formed in the skiving stage by joining several reels (with glue or tape) length-wise. It consists of reels of full lengths because of the indivisibility. The reels are arranged in non-increasing order of their widths. That is, the first/top reel has the maximum width and the last/bottom reel has the minimum width. The text on the left of Figure 1 denotes the indexes, widths and lengths of the reel types used. For example, the top reel belongs to type-1 and has width 1215 and length 4500. The length of the reel-pyramid is equal to the total length of the included reels; it is 26700 in Figure 1.

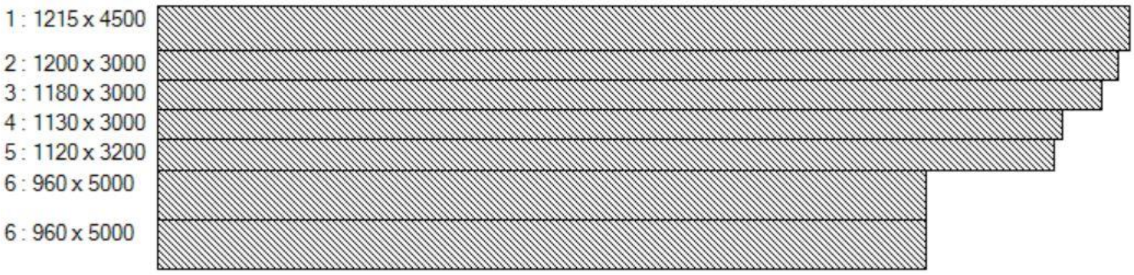


Figure 1. Reel-pyramid.

The roll-pyramid contains all required rolls that are arranged in several *roll-bars* (see Figure 2). Each roll-bar contains rolls of the same length. The roll-pyramid in Figure 2 contains three roll-bars. Each solid rectangle in a roll-bar represents a finished roll, the text in which one denotes the type and width of the roll. The roll-bars in the roll-pyramid are arranged in non-increasing order of their widths, where the first/top bar has the maximum width and the last/bottom bar has the minimum width. The lengths of the first and last roll-bars are both 10000, and that of the second roll-bar is 6000. The length of the roll-pyramid is 26000, which is smaller than that of the reel-pyramid in Figure 1 by 700.

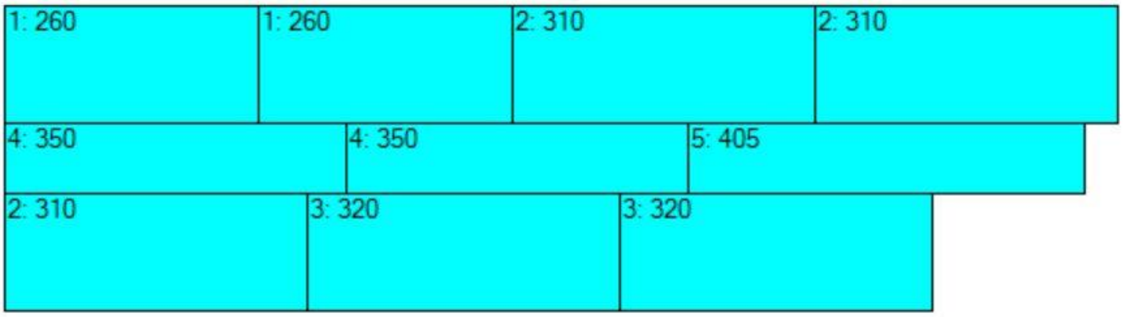


Figure 2. Roll-pyramid.

An ISCSP solution to the SCSP example is shown in Figure 3. It consists of the reel-pyramid of Figure 1 and the roll-pyramid of Figure 2. The roll-pyramid must fit entirely within the reel-pyramid. In the cutting stage, the reel-pyramid is split width-wise into order rolls. The cutting plan shown in Figure 3 contains three roll-bars. The first roll-bar contains four rolls (two type-1 rolls and two type-2 rolls) of length 10000; and the second roll-bar contains three rolls (two type-4 rolls and one type-5 roll) of length 6000. Some reels are used across successive roll-bars (third and fifth reels), and each of the other reels is used completely by a roll-bar.

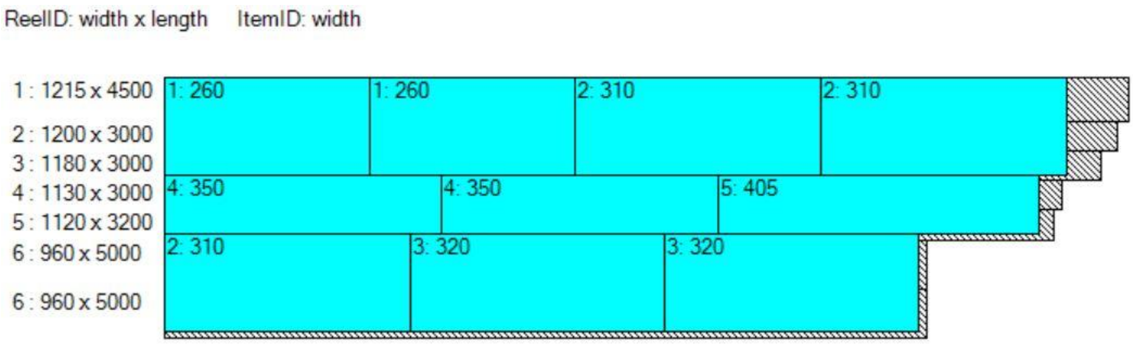


Figure 3. ISCSP cutting plan of the SCSP example.

To help set up a mathematical model for the DSCSP, we define *strip types*, which correspond to reel types. Recall that L_j is the length of a type-j reel and N_j is the number of type-j reels. Correspondingly, a type-j strip can be defined as a rectangle with width W_j and unit length (1 meter). With this definition, a type-j reel can be taken as L_j strips of type-j, and the total number of available type-j strips becomes $D_j = N_j L_j$ (previously referred to as the total length of type-j reels), $j \in J$.

The solution of the DSCSP is a cutting plan that contains a set of different *cutting patterns* (simply called *pattern* in the rest of paper). As shown in Figure 4, a pattern includes a strip-bar and a roll-bar. Only rolls of the same length are allowed to appear in a pattern. Both the strip-bar and the roll-bar should have the same length. The roll-bar must fit entirely within the strip-bar. In Figure 4 (a), each tuple above the picture represents the type, width, and number of strips. The strip-bar consists of three strip types. The widths and numbers of the strip types are 1215×4000 (width × number), 1200×3000 and 1180×3000, respectively. The total number of strips is 10000, which is equal to the length of the rolls. The text in Figure 4 (b) shows that the roll-bar contains two type-1 rolls and two type-2 rolls. The width of a type-1 roll is 260 and that of a type-2 roll is 310. The total width of the rolls (1140) is equal to that of the roll-bar, and it does not exceed the minimum width (1180) of the strips.

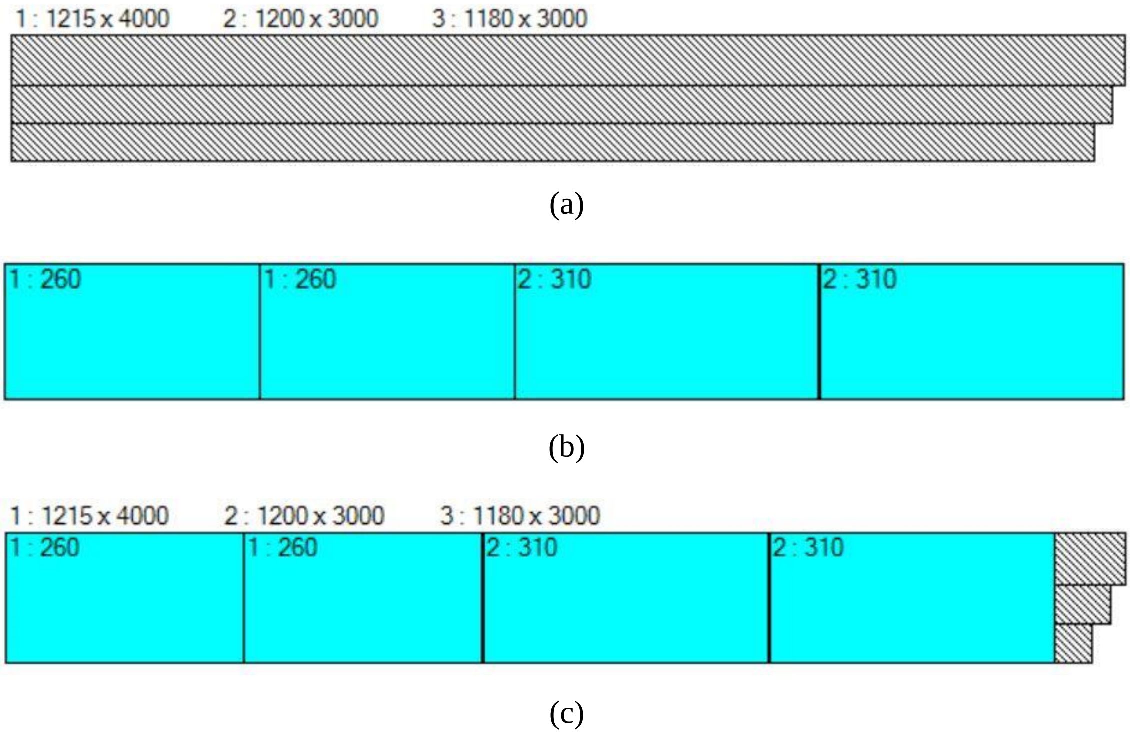


Figure 4. Pattern structure. (a) Strip-bar. (b) Roll-bar. (c) Pattern.

There are seven strip types and five roll types in the SCSP example. In each pattern, an edge trim of 10 mm of the strip bar is required for cutting out the last roll on the roll-bar. This means that the width of the thinnest strip in a strip-bar must be larger than the width of the related roll-bar by at least 10 mm.

A cutting plan of the example is shown in Figure 5. It contains three patterns. The first two patterns contain rolls of length 10000, and the third contains rolls of length 6000. For each pattern, "ID : Fre" on the left denotes the index and frequency of the pattern, and "Strip ID : Width × Number" above the pattern denotes the data (type, width, number) of each strip type in the pattern. All patterns have frequency 1. The first pattern contains two type-1 rolls and two type-2 rolls. It uses 4000 type-1 strips, 3000 type-2 strips, and 3000 type-3 strips. The other two patterns can be interpreted similarly.

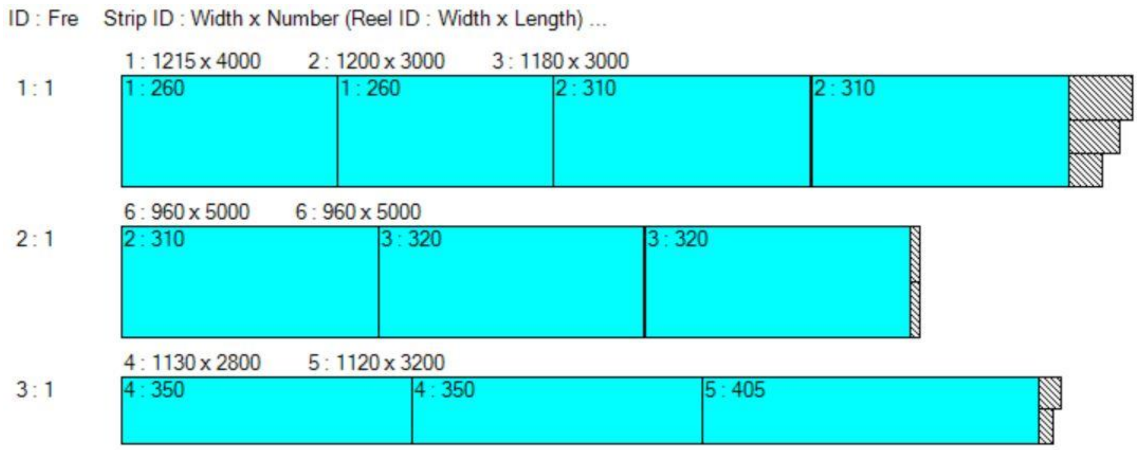


Figure 5. DSCSP cutting plan of the SCSP example.

Figure 6 shows the DSCSP solution that is obtained by rearranging the patterns of the DSCSP solution in Figure 5 in non-increasing order of their roll-bar widths and drawing the picture without the space between the patterns. In this manner, the DSCSP solution can also be seen as consisting of a reel-pyramid and a roll-pyramid. The reel-pyramid is different from that of the ISCSP solution (Figure 3) in the following two aspects:

(1) The latter consists of full reels. The former can include sub-reels, for example, the type-1 reel is divided into two sub-reels; one with length 4000 is used as the top reel of the reel-pyramid and the other with length 500 is retained in inventory.

(2) In the latter, the possible leftover in the bottom reel is taken as waste; for example, the bottom reel in Figure 3 contains a leftover (width 960 and length 700) that is taken as waste. In the former, no length-wise leftover is taken as waste, because the total length of the reel-pyramid is equal to that of the roll-pyramid.

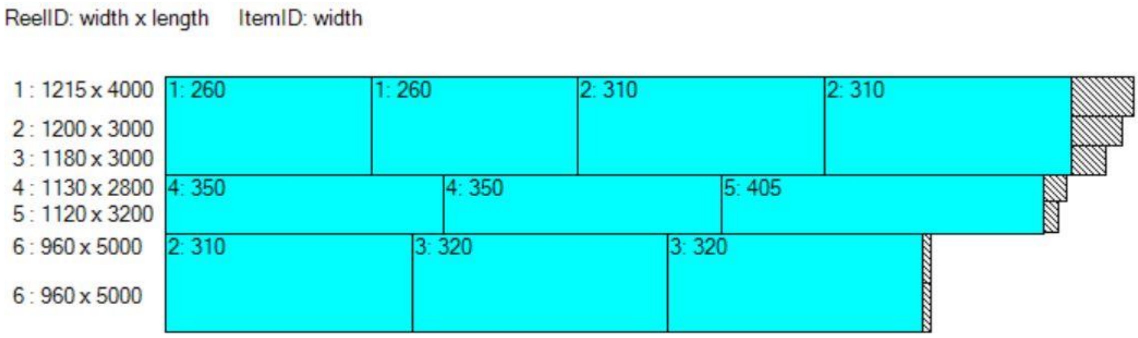


Figure 6. DSCSP cutting plan equivalent to that of Figure 5

In this paper, we first set up one Integer Programming (IP) model for the DSCSP and one IP model for the sub-problem of the ISCSP, respectively. To obtain efficient solutions for the SCSPs, a sequential value correction procedure (SVC) combined with the Two IP models developed (SVCTIP) is proposed. The effectiveness of the SVCTIP is demonstrated through computational tests.

The remainder of this paper is organized as follows. Section 2 presents a literature review. Section 3 describes the relationships between the DSCSP and ISCSP solutions. Sections 4 and 5 respectively investigate the DSCSP and ISCSP. To make a comparison with the SVCTIP algorithms proposed in this paper, a well-known Sequential Heuristic Procedure (SHP) algorithm is presented in Section 6 to provide solutions to the DSCSP and ISCSP. Section 7 presents the computational results. Finally, Section 8 presents the conclusions.

2. Literature review

When solving cutting stock problems (CSPs) in the paper industry, it is generally assumed that assembling original or intermediate objects to produce ordered items is not allowed. Here, original objects represent the stock objects that are available at the beginning of the cutting process and intermediate objects, denote those produced from the original objects during the cutting process and will be cut further to produce finished items.

Several algorithms use this assumption in solving CSPs in the paper industry, such as Correia et al. (2004), Chauhan et al. (2008), and Kallrath et al. (2014). Correia et al. (2004) discussed the roll and sheet cutting problem at a paper mill. First, the reels are split into auxiliary reels of smaller width. Then, the auxiliary reels are cut into finished rolls or sheets to satisfy customer orders. The authors proposed a solution procedure that includes three stages. The first stage enumerates all feasible and desirable auxiliary reels and cutting patterns. The second stage solves linear programming models over the cutting patterns generated in the first stage. The third stage uses a rounding procedure to obtain integer solutions from the often fractional solutions produced in the second stage.

Chauhan et al. (2008) discussed the roll assortment problem in a paper mill, in which huge reels produced on a cyclical basis on paper machines are cut into rolls of smaller size; these rolls are then sold directly or after splitting into finished products. A huge number of roll sizes would be required to cut all finished products without trim loss. It is necessary to inventory an assortment of rolls owing to the limited availability of storage space; as a result, trim loss is incurred. The assortment of rolls to inventory should be determined through optimization to reduce trim loss and other costs. The authors formulated the problem as a binary nonlinear programing model and presented two solution methods to solve it. The first is a branch and price algorithm based on column generation and a fast pricing heuristic. The second is a marginal cost heuristic.

Kallrath et al. (2014) developed new column generation approaches for two cases in the paper industry: CSPs minimizing underproduction and CSPs with reels of different widths and availability.

A few algorithms allow the assembly of intermediate objects width-wise to produce ordered items, such as Johnson et al. (1997) and Arbib and Marinelli (2005). Johnson et al. (1997) investigated the cutting and skiving stock problem (CSSP) in the paper industry, in which a two-stage process is used: first, stock reels are cut into finished rolls and small auxiliary rolls and second, auxiliary rolls are skived (glued side-by-side, with some overlap) width-wise to obtain additional finished rolls.

The obvious differences between the CSSP and the SCSP include the following. (1) A cutting pattern of the CSSP uses one reel, whereas that of the SCSP often uses multiple reels that are skived together. (2) A finished roll of the CSSP contains glue seam/seams along the length direction, whereas that of the SCSP includes glue seam/seams along the width direction. The total length of the glue seams in the CSSP is much larger than that in the SCSP because the roll length is often thousands of times the roll width. Subsequently, the cosmetic characteristic of the finished rolls is better in SCSP than in CSSP.

Arbib and Marinelli (2005) investigated a CSSP in a European plant devoted to the production of gear belts. The production process includes two stages. In the first stage, reels are cut into rectangular items. In the second stage, an item may be used directly or after sewing with another item (the skiving action). The authors presented an optimization model that integrates process optimization and inventory planning.

The skiving stock problem (SSP) is investigated in some papers (Zak, 2003; Martinovic and Scheithauer, 2015), in which several reels with specified availabilities are skived width-wise and possibly in different ways to produce a finished roll. The SSP and SCSP are also different. In the former, reels are skived width-wise, and in the latter, they are skived length-wise. In the former, a generated reel-pyramid is used to produce one finished roll, and in the latter, it is often cut into multiple finished rolls.

Although the cutting stock problem with usable leftovers has been addressed in the literature (Scheithauer, 1991; Cui and Yang, 2010; Cui et al., 2016), the approaches assume that leftovers generated in previous cutting processes cannot be assembled into a large object that can be used to produce items.

In summary, the SCSPs are new cutting stock problems proposed by a thermal and other fine papers manufacturer to Greycon (www.greycon.com), a company specializing in applying advanced mathematical techniques in the manufacturing industry worldwide. Although various approaches have been published for solving various cutting stock problems, no publication has been found, to our best knowledge, to solve the SCSPs because of their unique structures. This finding has motivated this research paper.

3. Symbols definition and relationships between DSCSP and ISCSP solutions

To illustrate the relationships between DSCSP and ISCSP solutions, three sets of mathematical symbols are defined in this Section. The first set is used for both DSCSP and ISCSP descriptions. The second set is used for ISCSP and the third set is used for DSCSP. The sizes of both reels and rolls take integral values.

Set 1: Symbols used in both DSCSP and ISCSP descriptions

n Number of reel types.

J Set of reel-type indexes, $J = \{1, \cdots, n\}$.

L_j Length of reels (in meters with abbreviation m), $j \in J$.

W_j Width of reels (in millimetres with abbreviation mm) $j \in J$.

N_j Number of reels, $j \in J$.

D_j The total length of type-j reels, $D_j = N_j L_j$, $j \in J$.

m Number of roll types.

I Set of roll indexes, $I = \{1, \cdots, m\}$.

l_i Length of rolls (in meters with abbreviation m), $i \in I$.

w_i Width of rolls (in millimetres with abbreviation mm), $i \in I$.

d_i Demand of rolls, $i \in I$

$\mathbf{Z}_0^+$ Set of non-negative integers.

Set 2: Symbols used in DSCSP description

K Number of feasible patterns. K could be a very large number theoretically.

P_k The k-th pattern $P_k = (a_{k1,\cdots,}a_{km}, b_{k1,\cdots,}b_{kn})$, where a_{ki} is the number of type-i rolls in P_k and b_{kj} is the number of type-j strips used by P_k, $i \in I, j \in J$, $k = 1, \cdots, K$.

s_k Total area of the strips in P_k, $s_k = \sum_{j \in J} b_{kj} W_j$, measured in $m \times mm$.

x_k Decision variable denoting the frequency (number of times to use) of P_k, $k=1,\cdots,K$.

Set 3: Symbols used in ISCSP description

τ Number of roll-bars.

h_t Length of the t-th roll-bar, $t=1,\cdots,\tau$.

ω_t Width of the t-th roll-bar, $t=1,\cdots,\tau$. $\omega_1 \geq \cdots \geq \omega_\tau$.

ψ_t Set of reel types with width larger than or equal to ω_t, $t=1,\cdots,\tau$.

χ_j Decision variable denoting the number of type-j reels in the reel-pyramid, $j \in J$.

When a DSCSP solution is known, the roll-pyramid in the corresponding ISCSP solution can be obtained by rearranging the roll-bars in the DSCSP solution in non-increasing order of their widths, for example, the roll-pyramid in Figure 2 (ISCSP) contains exactly the three roll-bars in Figure 5 (DSCSP). In other words, knowing a DSCSP solution, τ takes the value of $\sum_{k=1}^{K} x_k$ and other data ($h_t, \omega_t, \psi_t \,|\, t=1,\cdots,\tau$) of the roll-pyramid in the corresponding ISCSP solution are also known. The method for determining the reel-pyramid of the corresponding ISCSP solution will be presented later in Section 5.

4. DSCSP optimisation model and solution method

First, the IP model of the DSCSP is presented, and then the lower bound of this problem is investigated. Finally, an algorithm combining a Sequential Value Correction procedure with the IP model of DSCSP (SVCIP) is proposed.

4.1. DSCSP optimisation model

The DSCSP is formulated as the following integer linear programming model.

DSCSP model: $\min A = \sum_{k=1}^{K} s_k x_k$

$$\sum_{k=1}^{K} a_{ki} x_k \geq d_i, \quad i \in I \tag{1}$$

$$\sum_{k=1}^{K} b_{kj} x_k \leq D_j, \quad j \in J \tag{2}$$

$$x_k \in \mathbf{Z}_0^+, \quad k=1,\cdots,K \tag{3}$$

The objective is to minimize the total area of strips used. Constraint (1) ensures that the demand of each roll type is met. Constraint (2) ensures that the number of each strip type used should not exceed the supply bound.

The solution is optimal if all patterns are considered in solving the DSCSP model, possibly using an optimization solver. However, given the fact that the number of feasible patterns is huge and the difficulty in enumerating them, only a subset of the patterns is considered in solving the DSCSP model in this paper. The subset of the patterns is generated with the SVCIP algorithm which is presented in Section 4.3. Thus the SVCIP algorithm is heuristic.

4.2. A lower bound for the DSCSP

The Linear Relaxation (LR) of the DSCSP model can be obtained by replacing $x_k \in \mathbf{Z}_0^+$ with $x_k \geq 0$ in constraint (3). A lower bound of the DSCSP can be obtained by solving the LR using Column-Generation (CG). The following description is based on our implementation of the CG using the optimization solver CPLEX (called CG_CPLEX).

Let $\mathbf{\Omega}$ be the set of patterns considered. Initially $\mathbf{\Omega}$ contains m patterns generated as follows. A pseudo reel type with width W_{n+1}=$100 \times \max_{j \in J} \{W_j\}$ and infinite supply bound is used to generate the initial patterns. For the i-th $(i \in I)$ pattern, the roll-bar contains only a type-i roll and the strip-bar consists of l_i strips of type-$(n+1)$. In the initial LR solution, the frequency of the i-th pattern is d_i, $i \in I$. Because the strip width W_{n+1} is very large, the frequencies of these initial patterns will be zero in the final LR solution in order to minimize the total area of the reels used.

Introduce the following functions:

Solve()	CPLEX function solving the LR over the patterns in $\mathbf{\Omega}$.
GetDuals()	CPLEX function obtaining the duals $[\beta_1, \cdots, \beta_m, \pi_1, \cdots, \pi_n]$ of the LR solution.
GetPatLR()	Function generating a new pattern P_k using the duals. How it works will be described later in this section.

Based on the principle of the CG (Lübbecke and Desrosiers, 2005), the new pattern P_k is said to be *promising* if the following inequality holds.

$$s_k - \sum_{i=1}^{m} \beta_i a_{ki} - \sum_{j=1}^{n} \pi_j b_{kj} = \sum_{j=1}^{n} W_j b_{kj} - \sum_{i=1}^{m} \beta_i a_{ki} - \sum_{j=1}^{n} \pi_j b_{kj} < 0 \tag{4}$$

The CG_CPLEX solves the LR through the following iteration.

While TRUE

Solve().

GetDuals().

GetPatLR().

If P_k is not promising then break.

Add P_k to $\mathbf{\Omega}$.

When the iteration terminates, the lower bound of the total area of the reels used is determined as $A_{LB} = \sum_{k=1}^{K} s_k x_k^{LR}$, where x_k^{LR} is the frequency of P_k in the optimal LR solution. The upper bound of material utilization is defined as $U_{UB} = \left(\sum_{i=1}^{m} w_i l_i d_i\right) \Big/ A_{LB}$. The following paragraphs describe the method for designing the function GetPatLR() that generates P_k.

From inequality (4), we have

$$\sum_{j=1}^{n} \left(W_j - \pi_j\right) b_{kj} < \sum_{i=1}^{m} \beta_i a_{ki} \ ; \ \sum_{i=1}^{m} \beta_i a_{ki} \Big/ \sum_{j=1}^{n} \left(W_j - \pi_j\right) b_{kj} > 1$$

$\left(W_j - \pi_j\right)$ is referred to as the cost of a type-j strip, $j \in J$. Define the output of P_k as $o_{best} = \sum_{i=1}^{m} \beta_i a_{ki} \Big/ \sum_{j=1}^{n} \left(W_j - \pi_j\right) b_{kj}$. Then the iteration will continue if $o_{best} > 1$. Therefore, the pattern should be determined such that o_{best} is the maximum.

As shown in Figure 4, a pattern includes a strip-bar and a roll-bar. Both the strip-bar and the roll-bar have the same length. Define the width of a strip-bar as the minimum width of the included strips. The strip-bar/roll-bar length and strip-bar width should be determined before generating a pattern. The strip-bar length l should be one in the set of $\left(l_1, \cdots, l_m\right)$, and the strip-bar width W should be one in the set of $\left(W_1, \cdots, W_n\right)$.

Knowing the length l and width W of the strip-bar, the strips forming the strip bar can be determined inder to minimize the total cost $\sum_{j=1}^{n} \left(W_j - \pi_j\right) b_{kj}$ of the strips. Among the strip types with width not smaller than W, the first l strips with the minimum costs are selected to form the strip-bar. Subsequently, b_{kj}, the number of type-j strips used by P_k, is known, $j \in J$; s_k, the total area of the strips in P_k, is also known. It is obvious that $b_{kj} \le D_j$ because of the supply bound of the strip types.

Only rolls of the same length are allowed to appear in the roll bar. Let $S(l)$ be the set of rolls of length l. The roll-bar is determined by solving the following bounded knapsack problem.

$$\begin{gathered} u_k = \max\left(\sum\nolimits_{i \in S(l)} \beta_i a_{ki}\right) \\ \sum\nolimits_{i \in S(l)} w_i a_{ki} \le W;\ \ a_{ki} \in \mathbf{Z}_0^+ \ \text{ and }\ a_{ki} \le d_i,\ \ i \in S(l) \end{gathered} \tag{5}$$

The model can be solved by the dynamic programming algorithm such as the one presented in Kellerer et al. (2004). The algorithm has the all capacity property: Once the model is solved for the maximum strip width $W_{\max} = \max_{j \in J}\{W_j\}$, the solutions to all strip widths $(W_1, \cdots, W_n)$ are also known. The roll types not belonging to $S(l)$ do not appear in the roll-bar. After solving Model (5), a_{ki}, the number of type-i rolls in P_k, is known, $i \in I$.

Initially let the output of the optimal pattern P_k be $o_{best} = 0$. The function GetPatLR() for obtaining the optimal/best pattern consists of the following steps.

Algorithm GetPatLR():

Step 1. Perform steps 2–3 for each different strip-bar length l in $(l_1, \cdots, l_m)$.

Step 2. Solve Model (5) for $W = W_{\max}$.

Step 3. Perform Steps 3.1–3.3 for each different strip-bar width W in $(W_1, \cdots, W_n)$.

Step 3.1. Obtain the strips in the strip-bar (as described previously).

Step 3.2. Obtain the rolls in the roll-bar (from the results of Step 2).

Step 3.3. If the output of the current pattern is larger than o_{best}

Record the current pattern as the best one and update o_{best}.

Step 4. Return the best pattern.

4.3. Sequential value correction procedure combined with DSCSP model

Sequential value correction procedures have been widely used in the literature to solve cutting and packing problems (Song et al., 2006; Belov et al., 2008; Cui et al., 2015). We propose a sequential value correction procedure and combine it with the DSCSP model to solve the DSCSP. The following symbols are used to describe the proposed heuristic algorithm for DSCSP (SVCIP), where SVC denotes the sequential value correction technique used to generate multiple solutions and IP indicates that an integer programming model is solved to obtain possible improvement.

$G_{\max}$	Number of cutting plans to be generated ($G_{\max}$ = 100 by default).
G	Index of the current cutting plan.
A_0	Total strip area of the best cutting plan.
A	Total strip area of the current cutting plan.
c_i	Value of a type-i roll, $i \in I$. It is necessary to assign a value for each roll to be used in generating each pattern.
r_i	Remaining demand of type-i rolls, $i \in I$.
R_j	Remaining supply of type-j strips, $j \in J$.
$\mathbf{C}$	$\mathbf{C}=[c_1,\cdots,c_m]$.
$\mathbf{r}$	$\mathbf{r}=[r_1,\cdots,r_m]$.
$\mathbf{R}$	$\mathbf{R}=[R_1,\cdots,R_n]$.

The SVCIP is described by the algorithm below.

Algorithm SVCIP:

Step 1. Let $A_0 = +\infty$ and $G=0$. Let $c_i = w_i$, $i = 1,\cdots,m$.

Step 2. Let $G \leftarrow G+1$. If $G=G_{\max}$ then go to Step 7.

Step 3. Let $r_i = d_i, i = 1,\cdots,m$. Let $R_j = D_j$, $j = 1,\cdots,n$. Let $k = 1$ and $A = 0$.

Step 4. While there exist remaining rolls, perform Steps 4.1–4.3.

Step 4.1. Call $\text{GetPat}(\mathbf{R},\mathbf{r},\mathbf{C})$ to determine pattern P_k and its frequency x_k.

Step 4.2. Add pattern P_k to the current cutting plan. Let $r_i \leftarrow r_i - a_{ki}x_k$ to update the remaining demands, $i = 1,\cdots,m$. Let $R_j \leftarrow R_j - b_{kj}x_k$ to update the remaining strips, $j = 1,\cdots,n$. Let $A \leftarrow A + \sum_{j=1}^{n} b_{kj}W_j x_k$.

Step 4.3. Call $\text{AdjustVal}(P_k)$ to update the roll values c_i for all $i \in I$.

Step 5. If $A < A_0$, then let $A_0 = A$, and record the current cutting plan as the best one.

Step 6. Go to Step 2.

Step 7. Solve the DSCSP model over all patterns generated in Steps 2–6. Update the best cutting plan if an improvement is obtained.

Step 8. Output the best cutting plan.

The roll values are initialized to their widths in Step 1. Then the SVCIP generates $G_{\max}$ cutting plans to select the best one (Step 2).

Steps 3–4 determine the current cutting plan. The remaining demands of the rolls and the remaining supplies of the strips are initialized to the original ones in Step 3. The patterns in the current cutting plan are generated sequentially by repeating Step 4. The current pattern and its frequency are determined in Step 4.1 by calling the function $\mathrm{GetPat}(\mathbf{R},\mathbf{r},\mathbf{C})$, which is described later. In Step 4.2, the current pattern is added to the current cutting plan, and the remaining roll demands and strip supplies are updated correspondingly. In Step 4.3, the roll values are adjusted by calling the function $\mathrm{AdjustVal}(P_k)$, which is described later. In Step 5, the current cutting plan is recorded as the best one whenever a reduction in the total strip area is obtained.

In Step 7, all different patterns generated in the previous steps are put together, and an optimization solver is used to solve the DSCSP model to check if a better cutting plan can be found compared to the best one found in previous steps. In solving the DSCSP model with many patterns, a time limit $T_2^{\max}$ ($T_2^{\max}=+\infty$ in default) may be used to avoid long computing time. When the size of the pattern set is too large, Step 7 may fail to obtain an improved solution.

The function $\mathrm{AdjustVal}(P_k)$ is called in Step 4.3 of SVCIP to adjust the roll values $\mathbf{C}=[c_1,\cdots,c_m]$. This is useful to diversify the cutting plans, because different roll values are used in generating the patterns. The following symbols are used to describe the function.

U_k	Material utilization of P_k, $U_k=\left(\sum_{i=1}^{m}a_{ki}w_il_i\right)\Big/\left(\sum_{j=1}^{n}b_{kj}W_j\right)$.
σ	Real number, $\sigma>0$ and $\sigma=0.3$ by default.
p	Real number, $p\geq 1$ and $p=1.03$ by default.

The function $\mathrm{AdjustVal}(P_k)$ adjusts the roll values using the following formula.

$$c_i=g_1c_i+g_2w_i^p/U_k\text{, where }g_2=\sigma a_{ki}x_k/d_k\text{ and }g_1=1-g_2$$

The formula is similar to other ones previously used in the literature, such as the ones in the works of Belov and Scheithauer (2007) and Cui and Tang (2014). The main idea of how the formula works is the following: (1) the information of the previous patterns is considered by the first term g_1c_i and that of the current pattern by the second term $g_2w_i^p/U_k$; (2) in the second term, w_i^p with $p>1$ indicates that rolls with larger width are given priority; and (3) $1/U_k$ in the second term denotes that the rolls in the current pattern are given priority when they lead to poor material utilization.

The function $\text{GetPat}(\mathbf{R},\mathbf{r},\mathbf{C})$ is called in Step 4.1 of the SVCIP to determine the current pattern P_k and its frequency x_k, where $\mathbf{R}=[R_1,\cdots,R_n]$ denotes the remaining strips; $\mathbf{r}=[r_1,\cdots,r_m]$ denotes the remaining rolls; and $\mathbf{C}=[c_1,\cdots,c_m]$ denotes the roll values. The function GetPat(R,r,C) is similar to the function GetPatLR() (see section 4.2), where we only change the values used by the method:

(1) Replace β_i with c_i, $i\in I$.

(2) Replace $(W_j-\pi_j)$ with W_j, $j\in J$.

(3) Replace D_j with R_j in determining the strip-bar, $j\in J$.

(4) Replace d_i with r_i in Model (5), $i\in S(l)$.

The frequency of P_k is

$$x_k=\min\left\{\min\left[\lfloor r_i/a_{ki}\rfloor \middle| a_{ki}>0\wedge i\in I\right],\min\left[\lfloor R_j/b_{kj}\rfloor \middle| b_{kj}>0\wedge j\in J\right]\right\}$$

5. ISCSP optimisation model and solution method

Given a known roll-pyramid in the ISCSP solution, the corresponding reel-pyramid can be optimally obtained by solving the following integer linear programming *ISCSP model*.

$$\min z_{ISCSP}=\sum_{j=1}^{n}W_jL_j\chi_j \tag{6}$$

$$\sum_{j\in\Psi(i)}L_j\chi_j\ge\sum_{k=1}^{i}h_i,\quad i=1,\cdots,\tau \tag{7}$$

$$\chi_j\in\mathbf{Z}_0^+ \text{ and } \chi_j\le N_j,\quad j\in J \tag{8}$$

The objective is to minimize the total area of the reels used. Constraint (7) guarantees that the reel-pyramid will completely cover the roll-pyramid. Constraint (8) guarantees that the number of reels used does not exceed the supply.

In the ISCSP, a reel can be either unused or used completely. With the ISCSP model developed in this section and the relationship between ISCSP and DSCSP solutions illustrated in Section 3, a straightforward algorithm for the ISCSP can be obtained as follows.

For each DSCSP cutting plan generated by the SVCIP algorithm, first rearrange the roll-bars to obtain a roll-pyramid, then solve the ISCSP model to obtain the reel-pyramid, finally combine the roll-pyramid and reel-pyramid to obtain an ISCSP solution. Among the ISCSP solutions generated, the one with the minimum total reel area is selected as the best solution.

This algorithm is referred to as the SVCTIP, where TIP refers to Two Integer Programming DSCSP and ISCSP models used in the solution process. The algorithm can generate both the ISCSP and DSCSP solutions.

To improve the time efficiency of the SVCTIP, the ISCSP model is not solved for all of the DSCSP solutions generated in Section 4.3 (including the possible one in Step 7 of algorithm SVCIP). The ISCSP model is only solved when the total area of consumed reels in the DSCSP solution, denoted by A, satisfies $A \le \xi A_0$, where A_0 is the total strip area of the best DSCSP solution and $\xi = 1.005$ in default. Furthermore, the ISCSP model is only solved for different roll-pyramids, because identical roll-pyramids will lead to the same ISCSP solution.

6. Sequential heuristic procedure (SHP) for the DSCSP and the ISCSP

The most widely used SHP in solving cutting and packing problems (Suliman, 2006) can be combined with the two IP models proposed in this paper to solve the two SCSPs. We implemented the SHP algorithm as follows:

Algorithm SHP:

Step 1. Let $c_i = w_i$, $r_i = d_i$, $i \in I$. Let $R_j = D_j$, $j \in J$. Let $k = 1$.

Step 2. While there exist remaining rolls, perform Steps 2.1–2.2.

Step 2.1. Call $\text{GetPat}(\mathbf{R}, \mathbf{r}, \mathbf{C})$ to determine pattern P_k and its frequency x_k.

Step 2.2. Add P_k to the cutting plan. Let $r_i \leftarrow r_i - a_{ki} x_k$ to update the remaining rolls, $i \in I$. Let $R_j \leftarrow R_j - b_{kj} x_k$ to update the remaining strips, $j \in J$. Let $k \leftarrow k + 1$.

Step 3. Output the DSCSP solution. Obtain the corresponding ISCSP solution by solving the ISCSP model.

Step 1 of the SHP sets the roll values to be their widths and initializes the remaining demands of the rolls and the remaining supplies of the strips. The patterns in the cutting plan are generated sequentially by repeating Step 2. Step 3 outputs the DSCSP solution and obtains the corresponding ISCSP solution by solving the ISCSP model.

The difference between the SHP algorithm and the SVCTIP is that the former generates only one cutting plan and the latter considers multiple cutting plans through adjusting the item values.

7. Computational results

The SVCTIP was coded in C# and executed on a Dell computer (Inspiron 3847, Intel Core i5-4440 3.3 GHz CPU, 8 GB RAM). Version 12.5 of CPLEX was used as the optimization solver to solve the DSCSP and ISCSP models.

The discussions of the computational results are divided into three subsections. First, 20 random instances are generated and described in Section 7.1. The effects of parameter values on the results of SVCTIP algorithm for these instances are also presented in Section 7.1. Second, the computational results of the SVCTIP and SHP for these instances are presented and compared in Section 7.2. Finally, the computational result for an industrial case study is given in Section 7.3. Default parameter values ($G_{\max} = 100$, σ= 0.3, $p = 1.03$, $\xi = 1.005$) are chosen as the result of Section 7.1 and are used in the following two sub-sections. Both the random instances and the detailed computational results are available from the supplementary file. The computation times are in seconds. We define the material utilization of a solution as follows:

$$Material\ utilization = \frac{total\ area\ of\ rolls\ required}{total\ area\ of\ reels\ used} \times 100$$

The following symbols are used:

U_{UB} Upper bound of material utilization (see Section 4.2).

U_{DSC} Material utilization of the DSCSP solution.

U_{ISC} Material utilization of the ISCSP solution

7.1. Effect of parameter values over 20 random instances

Twenty instances were generated randomly. They are used because benchmark instances are not available. The rolls data of an instance are generated according to the following method, which is chosen according to the correspondence with the Greycon to make the instances more practical. The number of roll types m is in [10, 20]. Roll width w_i is in [150, 650] and must be a multiple of 5; Roll length l_i assumes one of the three values 5000, 7500 and 10000; Roll demand d_i is in [1, 8]; $i = 1, \cdots, m$.

Let $S_{items} = \sum_{i=1}^{m} w_i l_i d_i$. The reels data of an instance are generated according to the following method. First generate two random reals γ_1 in [1.3, 1.7] and γ_2 in [0.4, 0.6]. Then repeatedly generate large reel types until the total area of the large reels reaches or exceeds $\gamma_1 S_{items}$. Finally repeatedly generate small reel types until the total area of the small reels reaches or exceeds $\gamma_2 S_{items}$. For a particular reel type j, the length L_j is in [3000, 13000] and must be a multiple of 100; The supply bound N_j is in [1, 4]; the width W_j, being the multiple of 5, is in [650, 1400] for a large type and in [400, 645] for a small type; $j = 1, \cdots, n$. The number of reel types n is a dependent variable. Some features of the generated instances are listed in Table 3.

Table 3. Some features of the 20 random instances

ID	m	n	S_{items}	ID	m	n	S_{items}
1	11	14	124400000	11	18	21	142350000
2	10	18	131612500	12	11	17	131850000
3	20	26	248012500	13	11	20	162987500
4	10	20	119650000	14	15	28	235900000
5	11	19	108362500	15	14	28	215487500
6	20	31	246975000	16	19	24	240050000
7	11	27	181962500	17	13	22	180250000
8	11	16	147162500	18	20	23	211975000
9	10	13	151175000	19	10	13	95300000
10	14	19	206150000	20	16	37	303275000

The instances are used to investigate the effect of four parameter values used in the SVCTIP algorithm. The first parameter is $G_{\max}$ that denotes the maximum number of cutting plans to generate in the SVCIP algorithm. The second and third parameters are σ and p used in the $\mathrm{AdjustVal}(P_k)$ of the SVCIP algorithm to diversify the cutting plans. The fourth parameter is ξ used in Section 5 to decide on the number of DSCSP solutions considered in solving the ISCSP model to improve the ISCSP solution. The default values of them are $G_{\max}$ = 100, $\sigma = 0.3$, $p = 1.03$, and $\xi = 1.005$. Their reasonableness will be verified by experimental analysis. In the following Table 4 to Table 7, all the data inputs under columns "Av." are the average values of the results for the 20 instances; the average computation times for the ISCSPs are also reported in row "t_{ISC}". In addition, all the data inputs under columns "Dev." are the standard deviation of the results for the 20 instances.

We first evaluate the effect of the $G_{\max}$ value, with other parameters assuming default values. Table 4 shows the computational results. It is noted that the material utilization initially increases fast with $G_{\max}$. The increment slows down when $G_{\max} \geq 25$. It is negligible when $G_{\max} > 100$ (the default value).

Table 4. Effect of $G_{\max}$ values on the SCSP solutions

$G_{\max}$	1		5		25		50		100		500	
Index	Av.	Dev.	Av.	Dev.	Av.	Dev.	Av.	Dev.	Av.	Dev.	Av.	Dev.
$\boldsymbol{U_{DSC}}$	96.4	2.1	98.2	1.2	98.4	1.3	98.4	1.3	98.4	1.3	98.4	1.3
$\boldsymbol{U_{ISC}}$	95.0	2.3	96.6	1.7	96.8	1.5	96.9	1.5	96.9	1.5	96.9	1.5
$\boldsymbol{t_{ISC}}$	0.1	0.1	0.5	0.6	1.3	1.1	1.9	1.8	2.7	2.6	6.9	6.7

Secondly, we evaluate the effect of the σ value used in the $\mathrm{AdjustVal}(P_k)$ function, with other parameters assuming default values. Table 5 shows the computational results. It is seen that both U_{DSC} and U_{ISC} is not sensitive to the σ value when $\sigma \in [0.2,\ 0.5]$. The default $\sigma = 0.3$ is appropriate.

Table 5. Effect of σ values on the SCSP solutions

σ	**0.1**		**0.2**		**0.3**		**0.4**		**0.5**		**0.6**		**0.7**	
Index	Av.	Dev	Av.	Dev	Av.	Dev	Av.	Dev	Av.	Dev	Av.	Dev	Av.	Dev
$\boldsymbol{U_{DSC}}$	98.3	1.3	98.3	1.3	98.4	1.3	98.3	1.3	98.4	1.3	98.3	1.3	98.3	1.3
$\boldsymbol{U_{ISC}}$	96.8	1.5	96.9	1.5	96.9	1.5	96.8	1.5	96.8	1.6	96.7	1.7	96.7	1.6
$\boldsymbol{t_{ISC}}$	3.1	2.9	3.4	3.2	2.7	2.6	2.9	2.6	3.1	2.6	2.7	2.3	2.9	2.2

Thirdly, we evaluate the effect of the p value used in the $\mathrm{AdjustVal}(P_k)$ function, with other parameters assuming default values. Table 6 shows the computational results. It is seen that both U_{DSC} and U_{ISC} are not sensitive to the p value when $p \in [1.01,\ 1.10]$. The default $p = 1.03$ is appropriate.

Table 6. Effect of p values on the SCSP solutions

p	**1.00**		**1.01**		**1.03**		**1.05**		**1.07**		**1.10**		**1.20**	
Index	Av.	Dev	Av.	Dev	Av.	Dev	Av.	Dev	Av.	Dev	Av.	Dev	Av.	Dev
$\boldsymbol{U_{DSC}}$	98.2	1.4	98.3	1.3	98.4	1.3	98.3	1.3	98.3	1.3	98.3	1.3	98.2	1.3
$\boldsymbol{U_{ISC}}$	96.7	1.6	96.8	1.6	96.9	1.5	96.8	1.6	96.8	1.5	96.9	1.5	96.7	1.5
$\boldsymbol{t_{ISC}}$	3.2	3.3	3.1	2.9	2.7	2.6	2.9	2.2	2.8	2.2	2.7	3.2	2.8	2.6

Finally, we evaluate the effect of the ξ value, with other parameters assuming default values. As mentioned in Section 5, for a DSCSP solution to be considered in solving the ISCSP model to improve the ISCSP solution, its total area A of consumed reels must be smaller than ξA_0, where A_0 is the total area of reels consumed by the best DSCSP solution. Table 7 lists the computational results. The average computation time increases with ξ, because larger ξ values allow more DSCSP solutions to be considered. The average material utilization U_{ISC} initially increases with ξ; it reaches the maximum value 96.90% when $\xi = 1.005$ and does not vary when ξ further increases. Thus $\xi = 1.005$ is taken as the default value.

Table 7. Effect of ξ values on the ISCSP solution

ξ	**1.000**		**1.003**		**1.005**		**1.010**		$+\infty$	
Index	Av.	Dev	Av.	Dev	Av.	Dev	Av.	Dev	Av.	Dev
$\boldsymbol{U_{DSC}}$	98.4	1.3	98.4	1.3	98.4	1.3	98.4	1.3	98.4	1.3
$\boldsymbol{U_{ISC}}$	96.8	1.6	96.9	1.6	96.9	1.5	96.9	1.5	96.9	1.5
$\boldsymbol{t_{ISC}}$	0.9	0.6	2.1	2.1	2.7	2.6	4.4	3.4	5.5	3.9

7.2. Computational results for random instances

As mentioned in the last paragraph of Section 2, algorithms that can solve the SCSP have not been reported in the literature, thus we designed and programmed the most widely used SHP in solving these SCSPs (see Section 6). In this section, we compare our SVCTIP algorithm with the SHP algorithm using the 20 random instances generated in Section 7.1.

In Table 8, the computational results of the SVCTIP are summarized in the six columns with head SVCTIP, where $\Delta_1 = U_{UB} - U_{DSC}$, $\Delta_2 = U_{UB} - U_{ISC}$, and $\Delta_3 = U_{DSC} - U_{ISC}$. The two columns with head SHP show the results of the SHP, where U_{DSC}^{SHP} is the material utilization of the DSCSP solution, and U_{ISC}^{SHP} is that of the ISCSP solution. In the last two columns of the table, $\Delta_{DSC} = U_{DSC} - U_{DSC}^{SHP}$ denotes the difference between the material utilizations of the SVCTIP and SHP solutions to the DSCSP, and $\Delta_{ISC} = U_{ISC} - U_{ISC}^{SHP}$ denotes the difference between the material utilizations of the SVCTIP and SHP solutions to the ISCSP. The last row lists the average values. The objective value of a solution (total area of consumed reels) can be determined from the corresponding material utilization shown in Table 8 and the total area of the required rolls shown in Table 3. Take the DSCSP solution to Instance 1 as an example. The objective value is equal to $100 S_{Items} / U_{DSC}$ which is $100 \times 124400000 / 96.29 = 129193063$.

Table 8. Computational results

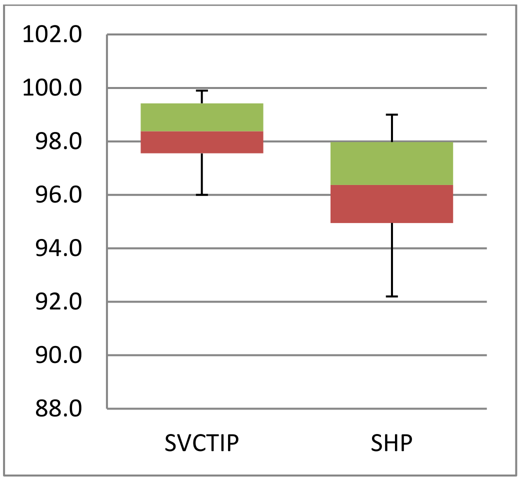
ID	U_{UB}	SVCTIP						SHP		Diff.	
		U_{DSC}	U_{ISC}	Δ_1	Δ_2	Δ_3	t_{ISC}	U_{DSC}^{SHP}	U_{ISC}^{SHP}	Δ_{DSC}	Δ_{ISC}
1	96.9	96.3	94.3	0.6	2.6	2.0	0.8	94.3	92.5	2.0	1.8
2	98.4	98.2	97.0	0.2	1.4	1.2	0.3	97.7	95.8	0.6	1.2
3	99.9	99.7	98.3	0.1	1.5	1.4	7.9	98.9	97.6	0.8	0.8
4	98.3	97.7	95.4	0.6	2.9	2.3	0.6	95.0	92.8	2.7	2.5
5	99.7	99.2	97.2	0.5	2.5	2.0	0.7	92.2	90.6	7.1	6.7
6	100.0	99.9	98.9	0.0	1.0	1.0	13.8	98.2	96.8	1.7	2.1
7	98.3	98.1	97.5	0.2	0.8	0.7	2.7	97.1	96.2	1.0	1.2
8	96.7	96.3	94.8	0.4	1.9	1.6	0.4	94.8	92.8	1.6	2.0
9	98.9	98.7	96.6	0.2	2.3	2.1	0.5	93.5	91.9	5.3	4.7
10	99.9	99.6	98.2	0.2	1.6	1.4	2.1	95.7	94.7	4.0	3.6
11	99.4	99.2	97.4	0.3	2.1	1.8	1.4	97.3	95.2	1.9	2.2
12	98.2	97.8	95.5	0.4	2.6	2.2	0.6	96.8	95.2	0.9	0.4
13	97.0	96.9	95.7	0.2	1.3	1.2	0.7	93.2	91.4	3.7	4.3
14	99.7	99.5	98.4	0.1	1.3	1.1	4.2	97.6	96.5	2.0	1.9
15	99.5	99.5	98.4	0.1	1.1	1.1	4.1	98.5	97.2	1.0	1.2
16	99.5	99.2	97.7	0.3	1.8	1.5	3.5	97.9	96.5	1.4	1.3
17	99.7	99.2	97.8	0.6	1.9	1.4	1.3	99.0	97.4	0.2	0.5
18	97.9	97.1	95.7	0.8	2.1	1.3	2.9	96.1	94.7	1.0	1.0
19	96.4	96.0	94.3	0.4	2.1	1.7	0.2	95.3	93.6	0.7	0.7
20	99.5	99.3	98.6	0.1	0.8	0.7	7.6	98.3	97.4	1.0	1.2
Av.	98.7	98.4	96.9	0.3	1.8	1.5	2.8	96.4	94.8	2.0	2.1

The computational results of Table 8 show that the DSCSP solutions are close to optimal, with the difference between the average material utilization of the DSCSP solutions and the upper bound being 0.3% (Δ_1 in the last row). The difference between the average material utilizations of the DSCSP and ISCSP solutions is 1.5% (Δ_3 in the last row), indicating that allowing the dividing of the stock reels is useful to improve material utilization. Although the average gap of the ISCSP solutions to upper bound of material utilization is 1.8% (Δ_2 in the last row), the average gap to the optimal solution may be far smaller, because the upper bound of material utilization obtained from solving the linear relaxation of the DSCSP model may be not tight.

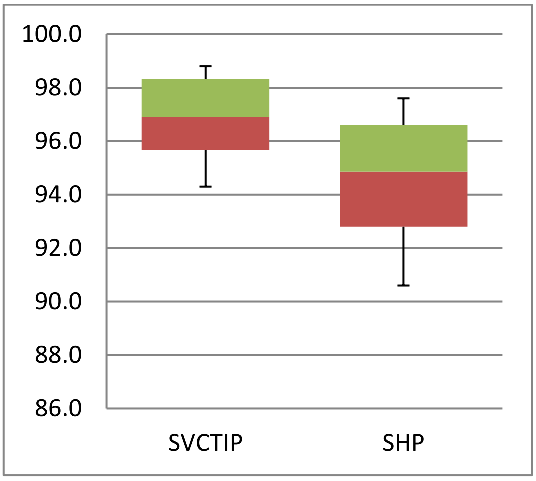
It is also noted that the average computation time of an instance with the SVCTIP is 2.8 seconds, which is fast enough from industry perspectives. The average running time of SHP is 0.1 seconds, which is much faster. However, SHP provides much worse solutions than those of the SVCTIP. The differences of the material utilizations of the

SHP from those of the SVCTIP are 2.0% for the DSCSP and 2.1% for the ISCSP in average.

We also provide a box plot in Figure 7 to assess the statistical significance of the observed differences from Table 8. In each box plot in Figure 7, the bottom and top of the box are the first and third quartiles, and the band inside each box is the mean. The ends of the whiskers represent the minimum and maximum of all of the data. For example, in Figure 7 (a) bottom and top of the box plot of the SVCTIP algorithm for the DSCSP are 97.5% and 99.4%. The corresponding mean value is 98.4% and the minimum and maximum values are 96.0% and 99.9%.



(a) Comparison for DSCSP



(b) Comparison for ISCSP

Figure 7. Statistical comparison of material utilization of SVCIP and SHP.

From Figure 7, it can be seen that none of the box plot reveals unusual features, such as gaps or outliers. In Figure 7 (a), the material utilization ratio is slightly less variable using the SVCTIP than using the SHP. Using the SVCTIP, the material utilization varies from 96.0% to 99.9 % (range = 3.9%) versus 92.2% to 99.0% (range = 6.8%) for the SHP. The mean material utilization is more telling - about 98.4% from SVCTIP versus 96.4% from SHP. The similar analysis can be applied to Figure 7 (b). It appears that the SVCTIP is more efficient than the SHP.

We now assess the statistical significance of the observed differences from Table 8 with more statistical tools. First, we apply the Shapiro-Wilk test for each set of results for each algorithm in Table 8. If they are normal, we are to apply test-t (since only two algorithms have been compared). If one of the distributions is not normal, we are to apply Wilcoxon U test. All the above mentioned statistical techniques are implemented using the open-source R (http://www.r-project.org). With the Shapiro-Wilk test for each set of results for each algorithm in Table 8, we achieve the Table 9.

Table 9. Statistical information of the computational results in Table 8

ID	SVCTIP		SHP	
	U_{DSC}	U_{ISC}	U_{DSC}^{SHP}	U_{ISC}^{SHP}
Average (mean)	98.4	96.9	96.4	94.8
Deviation	1.3	1.51	2.03	2.21
W	0.89	0.94	0.92	0.92
p-value	0.02	0.22	0.10	0.12

Form Table 9, we find that for an alpha level of 0.05, only the results for the DSCSP using SVCTIP in Table 8 is normal (under column "U_{DSC}"), since the corresponding *p*-values is 0.02, which is less than 0.05. All the other results in Table 8 are not normal, since all other *p*-values are greater than 0.05. Thus we need to apply Wilcoxon U test compare the SVCTIP and SHP for the DSCSP and ISCSP respectively.

We run the wilcox.test() function using the R Console statistical tool to make the Wilcoxon U test for the experimental results of SVCIP and SHP for the DSCSP. The following statistical information is achieved: p-value = 4.778e-05 for the alternative hypothesis that an experimental result of SVCIP for the DSCSP will be greater than a randomly selected value from the experimental results of SHP for the DSCSP. This alternative hypothesis holds since p-value << 0.05.

We also run the wilcox.test() function using the R Console statistical tool to make the Wilcoxon U test for the experimental results of SVCIP and SHP for the ISCSP. The following statistical information is achieved: p-value = 9.537e-07 for the alternative hypothesis that an experimental result of SVCIP for the ISCSP will be greater than a randomly selected value from the experimental results of SHP for the ISCSP. This alternative hypothesis holds since p-value < <0.05.

7.3. Computational result using an industrial case study

The industrial case study considers 25 reel/strip types and 14 roll types. An edge trim of 10 mm is required for the reels/strips. The data of the reel/strip types are shown in Table 10 and those of the roll types, in Table 11. The Greycon solution to this industrial case study was obtained through a correspondence with the company (Greycon's program that can solve the SCSP is not available to us). Its material utilization is 94.960%.

Table 10. Reel/strip data of the industrial case study (edge trim = 10 mm)

j	W_j	L_j	N_j	j	W_j	L_j	N_j	j	W_j	L_j	N_j
1	1360	9800	2	11	1060	5400	2	21	580	6000	1
2	1340	3400	2	12	1020	9200	3	22	560	10800	3
3	1320	3200	3	13	960	9800	1	23	500	11600	4
4	1300	3400	1	14	800	9600	1	24	440	12800	4
5	1300	4000	1	15	800	3800	4	25	400	5000	3
6	1280	8800	1	16	740	6200	1				
7	1220	10400	2	17	700	5400	1				
8	1200	5400	1	18	680	10800	3				
9	1120	8200	4	19	640	4600	2				
10	1100	7000	3	20	600	9000	2				

Table 11. Roll data of the industrial case study

i	w_i	l_i	d_i	i	w_i	l_i	d_i	i	w_i	l_i	d_i
1	160	10000	3	6	360	10000	1	11	575	10000	5
2	190	10000	1	7	405	10000	5	12	585	10000	2
3	300	10000	1	8	460	10000	1	13	620	10000	3
4	310	10000	3	9	470	10000	2	14	630	10000	8
5	350	10000	2	10	495	10000	2				

The ISCSP solution obtained from the SVCTIP is shown in Figure 8. Its material utilization is $U_{ISC} = 97.828\%$, which is higher than that of the Greycon solution by 2.868%. The solution for the DSCSP is shown in Figure 9. Its material utilization is $U_{DSC} = 98.374\%$, which is only 0.379% worse than the upper bound $U_{UB} = 98.753\%$. The Computation times for the ISCSP and DSCSP solutions are 1.69 and 0.27 s, respectively.

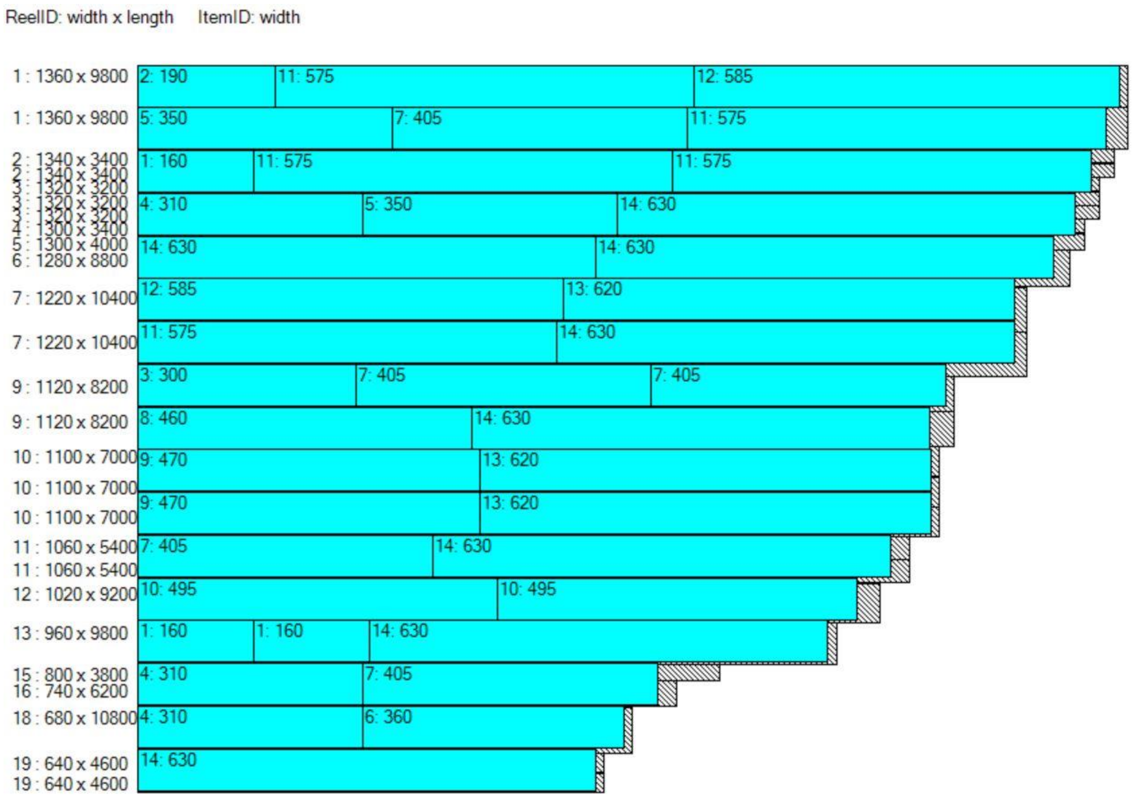


Figure 8. ISCSP solution to the industrial case study.

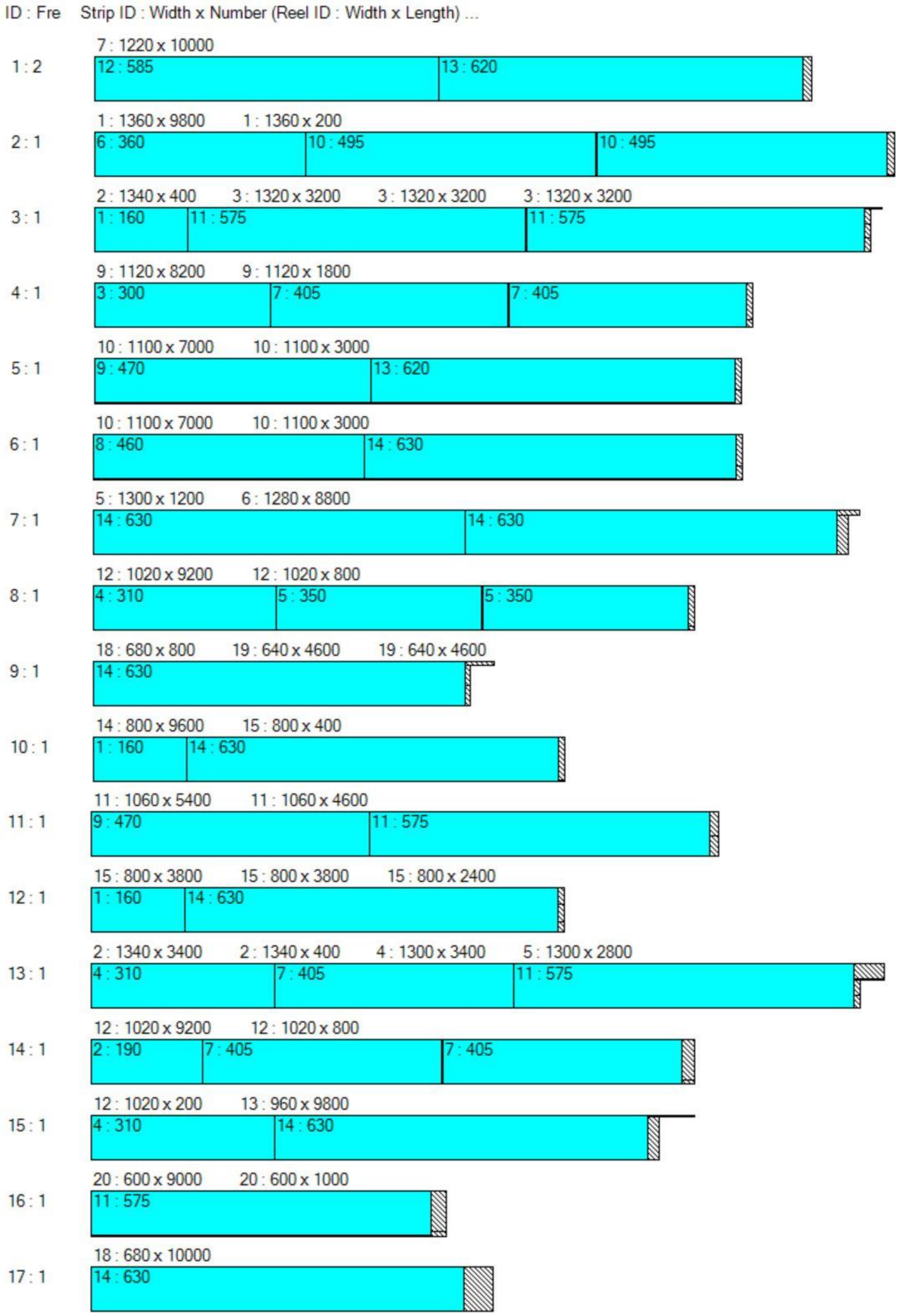


Figure 9. DSCSP solution to the industrial case study.

7. Conclusions

The SCSP appears in the paper and plastic film industries, in which a set of non-standard reels generated from previous cutting processes is used to produce finished rolls by the skiving and cutting process. First, reels are skived together length-wise to form a reel-pyramid (a polygon) and then, the reel-pyramid is cut into finished rolls of small widths. The cutting plan should be determined to minimize the total area of the reels used. Two sub-problems are investigated: the DSCSP in which the reels are divisible to form the reel-pyramid, and the ISCSP in which the reels are indivisible. Two IP models are developed for the two problems respectively. A sequential value correction procedure combined with the two IP models (SVCTIP) is proposed to solve the two SCSPs.

Three sets of computational tests are provided to evaluate the SVCTIP. In the first set of computational test, an industrial case study is solved with the SVCTIP and the computational results show that the material utilization of the ISCSP is 2.868% higher than that of the solution provided by the industry. This indicates that the SVCTIP can lead to better material utilization if it is used to design practical applications.

In the second set of computational test, 20 random instances are solved with the SVCTIP and the SHP, with the latter being a heuristic often used in solving cutting and packing problems. Three conclusions can be obtained from the test:

(1) The SVCTIP can yield close-to-optimal DSCSP solutions. The average gap of the DSCSP solutions to the upper bound of material utilization is only 0.31%.

(2) Allowing the dividing of the reels in forming the reel-pyramid is useful to reduce the total reel cost. For the 20 instances tested, the difference between the average material utilizations of the DSCSP and ISCSP solutions is 1.48%. In the DSCSP, it is often necessary to divide a stock reel. The produced leftover may be used by other patterns in the cutting plan or returned to inventory for future use. Subsequently, additional costs are incurred for dividing the reel and handling the leftovers. To make a reasonable decision on the divisibility of the stock reels, both the reel cost and additional costs should be considered.

(3) Compared with the SHP, using the SVCTIP can lead to significant improvement on material utilization. For the 20 instances tested, using the SVCTIP can increase the material utilization by more than 2%.

The third set of the computational test provides the systematic analysis of the effects of the four parameters in the SVCTIP on the computational results of the SCSPs. The conclusion is that the default values of the four parameters were set appropriately.

Although we have presented a SVCTIP and we believe that the ISCSP results are found close to their optimum, a tighter lower bound on the total reel area used could be explored in future research. Another research direction is to explore other research methods for solving the SCSP. For example, a column-generation based heuristic technique may be developed using similar techniques as described in Cui et al. (2015).

Greycon Company has informed us that they have improved their algorithm; however, the details of the improved algorithm are not available to the public. It is necessary to report our algorithm because the SCSP may be encountered more widely in the paper and plastic film industries, and no effective algorithm for solving this type of problem has been reported.

Acknowledgments

This research is part of Projects 71371058 and 61363026 supported by National Natural Science Foundation of China.

References

Arbib, C., Marinelli, F. 2005. Integrating process optimization and inventory planning in cutting-stock with skiving option: An optimization model and its application. European Journal of Operational Research 163 (3), 617–630

Arbib, C., Marinelli, F. Ventura, P. 2016. One-dimensional cutting stock with a limited number of open stacks: bounds and solutions from a new integer linear programming model. International Transactions in Operational Research 23 (1–2), 47–63

Beeker, K. H., Appa, G. 2015. A heuristic for the Minimum Score Separation Problem, a combinatorial problem associated with the cutting stock problem. Journal of the Operational Research Society 66 (4), 647–656.

Belov, G., G. Scheithauer. 2007. Setup and Open-Stacks Minimization in One-Dimensional Stock Cutting. INFORMS Journal on Computing 19 (1), 27–35

Belov, G., d. 2008. One-dimensional heuristics adapted for two-dimensional rectangular strip packing. Journal of the Operational Research Society 59 (6), 823–832.

Chauhan, S. S., Martel, A., Amour, S. D. 2008. Roll assortment optimization in a paper mill: An integer programming approach. Computers & Operations Research 35 (2), 614 – 627

Correia, M. H., Oliveira, J. F., Ferreira, J. S. 2004. Reel and sheet cutting at a paper mill. Computers & Operations Research 31 (8), 1223 – 1243

Cui, Y-P., Cui, Y., Tang, T. 2015. Sequential heuristic for the two-dimensional bin-packing problem. European Journal of Operational Research, 240 (1), 43–53.

Cui, Y, Cui, Y-P, Zhao, Z. 2015. Pattern set generation algorithm for the one-dimensional multiple stock sizes cutting stock problem. Engineering Optimization, 47 (9), 1289–1301.

Cui, Y., Song, X., Chen, Y., Cui, Y-P, 2016. New model and heuristic solution approach for one-dimensional cutting stock problem with usable leftovers. Journal of the Operational Research Society, accepted.

Cui, Y-P., Tang, T. 2014. Parallelized sequential value correction procedure for the one-dimensional cutting stock problem with multiple stock lengths. Engineering Optimization 46 (10), 1352–1368.

Cui, Y., Yang, Y. 2010. A heuristic for the one−dimensional cutting stock problem with usable leftover. European Journal of Operational Research, 204 (2), 245–250.

Garraffa, M., Salassa, F., Vancroonenburg, W., Berghe, G. V., Wauters, T. 2016. The one-dimensional cutting stock problem with sequence-dependent cut losses. International Transactions in Operational Research 23 (1–2), 5–24.

Johnson, M. P., Rennick, C., Zak, E. 1997. Skiving addition to the cutting stock problem in the paper industry. SIAM Review 39 (3), 472–483.

Kallrath, J., Rebennack, S., Kallrath, J. 2014. Solving real-world cutting stock-problems in the paper industry. Mathematical approaches, experience and challenges. European Journal of Operational Research 238 (1), 374–389

Kellerer, H., Pferschy, U., Pisinger, D. 2004. Knapsack Problems. Springer-Verlag, Berlin, 190–191.

Lübbecke, M., Desrosiers, J. 2005. Selected topics in column generation. Operations Research 53 (6), 1007–1023

Martinovic, J., Scheithauer, G. (2015). Integer linear programming models for the skiving stock problem. European Journal of Operational Research, doi: 10.1016/j.ejor.2015.11.005

Scheithauer, G. 1991. A note on handling residual lengths, Optimization 22, 461–466.

Song, X., Chu, C. B., Nie, Y. Y. 2006. An iterative sequential heuristic procedure to a real-life 1.5-dimensional cutting stock problem. European Journal of Operational Research 175 (3), 1870–1889

Stoyan, Y., Pankratov, A., Romanova, T. (2015). Cutting and packing problems for irregular objects with continuous rotations: mathematical modelling and non-linear optimization. Journal of the Operational Research Society, doi: 10.1057/jors.2015.94

Suliman, S. M. A. 2006. A sequential heuristic procedure for the two-dimensional cutting-stock problem. International Journal of Production Economics, 99 (1–2), 177–185

Wei, L., Qin, H., Cheang, B., Xu, X. 2016. An efficient intelligent search algorithm for the two-dimensional rectangular strip packing problem. International Transactions in Operational Research, 23 (1-2), 65–92

Zak, E. J. 2003. The skiving stock problem as a counterpart of the cutting stock problem. International Transactions in Operations Research 10 (6), 637–65